An algorithm for list decoding number field codes

Jean-François Biasse, Guillaume Quintin

An algorithm for list decoding number field codes

Jean-François Biasse
Department of Computer Science
University of Calgary
2500 University Drive NW
Calgary, Alberta, Canada T2N 1N4
Email: biasse@lix.polytechnique.fr

Guillaume Quintin
LIX
École Polytechnique
91128 Palaiseau, France
Email: quintin@lix.polytechnique.fr

***Abstract*—We present an algorithm for list decoding codewords of algebraic number field codes in polynomial time. This is the first explicit procedure for decoding number field codes whose construction were previously described by Lenstra [12] and Guruswami [8]. We rely on a new algorithm for computing the Hermite normal form of the basis of an $\mathcal{O}_K$-module due to Biasse and Fieker [2] where $\mathcal{O}_K$ is the ring of integers of a number field K.**

I. Introduction

Algorithms for list decoding Reed-Solomon codes, and their generalization the algebraic-geometric codes are now well understood. The codewords consist of sets of functions whose evaluation at a certain number of points are sent, thus allowing the receiver to retrieve them provided that the number of errors is manageable.

The idea behind algebraic-geometric codes can be adapted to define algebraic codes whose messages are encoded as a list of residues redundant enough to allow errors during the transmission. The Chinese Remainder codes (CRT codes) have been fairly studied by the community [10], [13]. The encoded messages are residues modulo $N := p_1, \cdots, p_n$ of numbers $m \leq K := p_1 \cdots p_k$ where $p_1 < p_2 < \cdots < p_n$ are prime numbers. They are encoded by using

$$\begin{array}{ccc} \mathbb{Z} & \longrightarrow & \mathbb{Z}/p_1 \times \cdots \times \mathbb{Z}/p_n \\ m & \longmapsto & (m \bmod p_1, \cdots, m \bmod p_n). \end{array}$$

Decoding algorithms for CRT codes were significantly improved to reach the same level of tolerance to errors as those for Reed-Solomon codes [3], [7], [10]. As algebraic-geometric codes are a generalization of Reed-Solomon codes, the idea arose that we could generalize the results for CRT codes to redundant residue codes based on number fields. Indeed, we can easily define an analogue of the CRT codes where a number field K plays the role of $\mathbb{Q}$ and its ring of integers $\mathcal{O}_K$ plays the role of $\mathbb{Z}$. Then, for prime ideals $\mathfrak{p}_1, \cdots, \mathfrak{p}_n$ such that $\mathcal{N}(\mathfrak{p}_1) < \cdots < \mathcal{N}(\mathfrak{p}_n)$, a message $m \in \mathcal{O}_K$ can be encoded by using

$$c: \begin{array}{ccc} \mathcal{O}_K & \longrightarrow & \mathcal{O}_K/\mathfrak{p}_1 \times \cdots \times \mathcal{O}_K/\mathfrak{p}_n \\ m & \longmapsto & (m \bmod \mathfrak{p}_1, \cdots, m \bmod \mathfrak{p}_n). \end{array}$$

The construction of good codes on number fields have been independently studied by Lenstra [12] and Guruswami [8]. They provided indications on how to chose number fields having good properties for the underlying codes. In particular, Guruswami [8] showed the existence of asymptotically good number field codes, that is a family $\mathcal{C}_i$ of $[n_i, k_i, d_i]_q$ codes of increasing block length with

$$\liminf \frac{k_i}{n_i} > 0 \text{ and } \liminf \frac{d_i}{n_i} > 0.$$

Neither of them could provide a decoding algorithm. In the concluding remarks of [8], Guruswami identifies the application of the decoding paradigm of [9], [11], [10] to number field codes as an open problem.

Contribution: The main contribution of this paper is to provide the first algorithm for decoding number field codes. We first show that a direct adaptation of an analogue of Coppersmith's theorem due to Cohn and Heninger [5] allows to follow the approach of Boneh [3] which does not allow to reach the Johnson bound. Then we adapt the decoding paradigm of [9, Chap. 7] to number field codes, by using methods for manipulating modules over the ring of integers of a number field recently described in [2] to achieve the Johnson bound.

Throughout this paper, we denote by K a number field of degree d, of discriminant Δ and of ring of integers $\mathcal{O}_K$. The prime ideals $(\mathfrak{p}_i)_{i\leq n}$ satisfy $\mathcal{N}(\mathfrak{p}_1) < \mathcal{N}(\mathfrak{p}_2) < \cdots < \mathcal{N}(\mathfrak{p}_n)$, and we define $N := \prod_{i\leq n} \mathcal{N}(\mathfrak{p}_i)$ and $B := \prod_{i\leq k} \mathcal{N}(\mathfrak{p}_i)^{1/d}$ for integers k, n such that $0 < k < n$. Before describing our algorithm in more details in the following sections, let us state the main result of the paper.

Theorem 1. *Let $\varepsilon > 0$, and a message $m \in \mathcal{O}_K$ satisfying $\|m\| \leq B$, then there is an algorithm that returns all the messages $m' \in \mathcal{O}_K$ such that $\|m'\| \leq B$ and that $c(m)$ and $c(m')$ have mutual agreement t satisfying*

$$t \geq \sqrt{k(n+\varepsilon)}.$$

This algorithm is polynomial in d , $\log(N)$, $1/\varepsilon$ and $\log|\Delta|$.

II. Generalities on number fields

Let K be a number field of degree d. It has $r_1 \leq d$ real embeddings $(\theta_i)_{i\leq r_1}$ and $2r_2$ complex embeddings $(\theta_i)_{r_1<i\leq r_1+2r_2}$ (coming as r_2 pairs of conjugates). The field K is isomorphic to $\mathcal{O}_K \otimes \mathbb{Q}$ where $\mathcal{O}_K$ denotes the ring of integers of K. We can embed K in

$$K_{\mathbb{R}} := K \otimes \mathbb{R} \simeq \mathbb{R}^{r_1} \times \mathbb{C}^{r_2},$$

and extend the θ_i's to $K_{\mathbb{R}}$. Let T_2 be the Hermitian form on $K_{\mathbb{R}}$ defined by

$$T_2(x, x') := \sum_i \theta_i(x)\overline{\theta_i}(x'),$$

and let $\|x\| := \sqrt{T_2(x, x)}$ be the corresponding L_2-norm. Let $(\alpha_i)_{i\leq d}$ be such that $\mathcal{O}_K = \oplus_i \mathbb{Z}\alpha_i$, then the discriminant of K is given by $\Delta = \det^2(T_2(\alpha_i, \alpha_j))$. The norm of an element $x \in K$ is defined by $\mathcal{N}(x) = \prod_i |\theta_i(x)|$.

We encode our messages with prime ideals of $\mathcal{O}_K$. However, for decoding, we need a more general notion of ideal, namely the fractional ideals of $\mathcal{O}_K$. A subset $\mathfrak{a} \subseteq K$ is said to be a fractional ideal if $\exists r \in \mathbb{Z}, r\mathfrak{a} \subseteq \mathcal{O}_K$. When a fractional ideal is contained in $\mathcal{O}_K$, we refer to it as an integral ideal.The sum and product of two fractional ideals of $\mathcal{O}_K$ is given by

$$\mathfrak{ab} = \{a_1b_1 + \cdots + a_lb_l \mid l \in \mathbb{N}, a_1, \cdots a_l \in \mathfrak{a}, b_1, \cdots b_l \in \mathfrak{b}\}$$

$$\mathfrak{a} + \mathfrak{b} = \{a + b \mid a \in \mathfrak{a}, b \in \mathfrak{b}\}.$$

Any non zero fractional ideal $\mathfrak{a}$ of $\mathcal{O}_K$ is invertible, that is there exists $\mathfrak{a}^{-1} := \{x \in K \mid x\mathfrak{a} \subseteq \mathcal{O}_K\}$ such that $\mathfrak{a}\mathfrak{a}^{-1} = \mathcal{O}_K$. The norm of integral ideals is given by $\mathcal{N}(I) := [\mathcal{O}_K : I]$, which extends to fractional ideals by $\mathcal{N}(I/J) := \mathcal{N}(I)/\mathcal{N}(J)$. The norm of a principal ideal agrees with the norm of its generator $\mathcal{N}(x\mathcal{O}_K) = |\mathcal{N}(x)|$.

In the following, we will study finitely generated sub $\mathcal{O}_K$-module of $\mathcal{O}_K[y]$. Let $M \subseteq K^l$ be a finitely generated $\mathcal{O}_K$-module. As in [4, Chap. 1], we say that $[(a_i), (\mathfrak{a}_i)]_{i\leq n}$, where $a_i \in K$ and $\mathfrak{a}_i$ is a fractional ideal of K, is a pseudo-basis for M if $M = \mathfrak{a}_1a_1 \oplus \cdots \oplus \mathfrak{a}_na_n$. We also call a pseudo-matrix representing M the matrix of the coefficients of the $(a_i)_{i\leq n}$ along with the ideals $\mathfrak{a}_i$. The algorithm [2, Alg.4] returns a pseudo-matrix representing M where the matrix of the $(a_i)_{i\leq n}$ has a triangular shape in polynomial time.

III. Decoding with Coppersmith's theorem

An analogue of Coppersmith's theorem was described by Cohn and Heninger in [5]. It was used to provide an elegant way of decoding Reed-Solomon codes, and the possibility to use it for breaking lattice- based cryptosystems in $\mathcal{O}_K$ modules was considered, although they concluded that it would not improve the state-of-the-art algorithms.

Theorem 2 (Cohn-Heninger)**.** *Let $f \in \mathcal{O}_K[X]$ be a monic polynomial of degree l, $0 < \beta \leq 1$, $\lambda_1, \cdots, \lambda_d > 0$ and $I \subsetneq \mathcal{O}_K$ an ideal. We can find in polynomial time all the $\omega \in \mathcal{O}_K$ such that $|\omega|_i := |\sigma_i(\omega)| \leq \lambda_i$ and*

$$\mathcal{N}(\gcd(f(\omega)\mathcal{O}_K, I) > \mathcal{N}(I)^\beta,$$

provided that the λ_i satisfy $\prod_i \lambda_i < (2+o(1))^{-d^2/2}\mathcal{N}(I)^{\beta^2/l}$.

Although not mentioned in [5], a straightforward adaptation of Theorem 2 with $\beta := \sqrt{\frac{\sum_{i\leq k} \log \mathcal{N}(\mathfrak{p}_i)}{\sum_{i\leq n} \log \mathcal{N}(\mathfrak{p}_i)}}$ where $0 < k < n$, $I := \prod_{i\leq n} \mathfrak{p}_i$ and $\forall i, \lambda_i := \prod_{i\leq k} \mathcal{N}(\mathfrak{p}_i)^{1/d}$ provides a polynomial time algorithm for decoding number field codes.

Theorem 3. *Let $(r_1, \cdots, r_n) \in \mathcal{O}_K^n$ and $m \in \mathcal{O}_K$ satisfying $\forall i,\ m = r_i \bmod \mathfrak{p}_i$, then Theorem 2 applied to $f(\omega) := \omega - m$ allows to return in polynomial time a list of $m' \in \mathcal{O}_K$ with $\mathcal{N}(m') \leq \prod_{i\leq k} \mathcal{N}(\mathfrak{p}_i)$ that differ from m in at most e places where*

$$e < n - \sqrt{kn\frac{\log \mathcal{N}(\mathfrak{p}_n)}{\log \mathcal{N}(\mathfrak{p}_1)}}.$$

In the rest of the paper, we present a method based on Guruswami's general framework for residue codes [9] that allows us to get rid in the dependency in $\frac{\log \mathcal{N}(\mathfrak{p}_n)}{\log \mathcal{N}(\mathfrak{p}_1)}$ in the decoding bound thus reaching the Johnson bound.

IV. Johnson-type bound for number fields codes

A Johnson-type bound is a positive number J depending on the distance, the blocklength and the cardinalities of the alphabets constituting the code. It garanties that a "small" number of codewords are in any sphere of radius J. By "small" number, we mean a number of codewords which is linear in the code blocklength and the dimension of the code. In our case, the Johnson-type bound for number fields codes depends only on the code blocklength and its minimal distance, and "small" means polynomial in $\sum_{i=1}^n \log \mathcal{N}(\mathfrak{p}_i)$.

The Johnson-type bound of [9, Section 7.6.1] remains valid for number field codes. For any prime ideal $\mathfrak{p} \subset \mathcal{O}_K$, the quotient $\mathcal{O}_K/\mathfrak{p}$ is a finite field. Thus the i'th symbol of a codeword comes from an alphabet of size $\mathcal{N}(\mathfrak{p}_i) = |\mathcal{O}_K/\mathfrak{p}_i|$ and [9, Th. 7.10] can be applied. Let t be the least positive integer such that $\prod_{i=1}^t \mathcal{N}(\mathfrak{p}_i) > \left(\frac{2B}{d}\right)^d$, where $d = [K : \mathbb{Q}]$ and let $T = \prod_{i=1}^t \mathcal{N}(\mathfrak{p}_i)$. Then, by [8, Lem. 12], the minimal hamming distance of the number fields code is at least $n-t+1$. Using [9, Th. 7.10], we can show that for a given message and $\varepsilon > 0$, only a "small" number of codewords satisfy

$$\sum_{i=1}^n a_i > \sqrt{(t+\varepsilon)n}, \tag{1}$$

where $a_i = 1$ if the codeword and the message agree at the i-th position, $a_i = 0$ otherwise. Thus, if our list decoding algorithm returns all the codewords having at most $n - \sqrt{(t+\varepsilon)n}$ errors then this number is garanteed to be "small". Therefore, the Johnson bound appears to be a good objective for our algorithm. Note that we would derive a different bound by using weighted distances. In particular, by using the log-weighted hamming distance i.e. $d(x, y) = \sum_{i:x\neq y \bmod \mathfrak{p}_i} \log \mathcal{N}(\mathfrak{p}_i)$, the condition would be $\sum_{i=1}^n a_i \log \mathcal{N}(\mathfrak{p}_i) > \sqrt{(\log T + \varepsilon)\log N}$.

V. General description of the algorithm

In this section, we give a high-level description of our decoding algorithm. We follow the approach of the general framework described in [9], making the arrangements required in our context. Our code is the set of $m \in \mathcal{O}_K$ such that

$\|m\| \leq B$ where $B = \prod_{i\leq k} \mathcal{N}(\mathfrak{p}_i)^{1/d}$. We also define $N := \prod_{i\leq n} \mathcal{N}(\mathfrak{p}_i)$. A codeword m is encoded via

$$\begin{array}{ccc} \mathcal{O}_K & \longrightarrow & \mathcal{O}_K/\mathfrak{p}_1 \times \cdots \times \mathcal{O}_K/\mathfrak{p}_n \\ m & \longmapsto & (m \bmod \mathfrak{p}_1, \cdots, m \bmod \mathfrak{p}_n). \end{array}$$

Let $z_1, \cdots, z_n$ be non-negative real numbers, and let Z be a parameter. In this section, as well as in Section VI and VII, we assume that the z_i are integers. We assume that we received a vector $(r_1, \cdots, r_n) \in \prod_i \mathcal{O}_K/\mathfrak{p}_i$. We wish to retrieve all the codewords m such that $\sum_i a_i z_i > Z$ where $a_i = 1$ if $m \bmod \mathfrak{p}_i = r_i$ and 0 otherwise (we say that m and $(r_i)_{i\leq n}$ have weighted agreement Z).

We find the codewords m with desired weighted agreement by computing roots of a polynomial $c \in \mathcal{O}_K[y]$ that satisfies

$$\|m\| \leq B \Longrightarrow \|c(m)\| < F, \tag{2}$$

for an appropriate bound F. We choose the polynomial c satisfying (2) in the ideal $\prod_{i\leq n} J_i^{z_i} \subseteq \mathcal{O}_K[y]$ where

$$J_i = \{a(y)(y - r_i) + p \cdot b(y) \mid a, b \in \mathcal{O}_K[y] \text{ and } p \in \mathfrak{p}_i\}.$$

With such a choice of a polynomial, we necessarily have $c(m) \in \prod_i \mathfrak{p}_i^{z_i a_i}$, where $a_i = 1$ if $c(m) \bmod \mathfrak{p}_i = r_i$, 0 otherwise. In particular, if $c(m) \neq 0$ then $\mathcal{N}(c(m)) \geq \prod_i \mathcal{N}(\mathfrak{p}_i)^{z_i a_i}$. In addition, we know, from the inequality beween arithmetic and geometric mean, that $\|c(m)\| \geq \sqrt{d}\mathcal{N}(c(m))^{1/d}$. We thus know that if the weighted agreement satisfies

$$\sum_{i\leq n} a_i z_i \log \mathcal{N}(\mathfrak{p}_i) > -\frac{d}{2}\log(d) + d\log(F), \tag{3}$$

which in turns implies $\sqrt{d}\left(\prod_i \mathcal{N}(\mathfrak{p}_i)^{z_i a_i}\right)^{1/d} > F$, then $c(m)$ has to be zero, since otherwise it would contradict (2).

Algorithm 1 Decoding algorithm

Require: $\mathcal{O}_K$, $z_1, \cdots, z_n$, B, Z, $r_1, \cdots, r_n \in \prod_i \mathcal{O}_K/\mathfrak{p}_i$.
Ensure: All m such that $\sum_i a_i z_i > Z$.
1: Compute l and F.
2: Find $c \in \prod_{i\leq n} J_i^{z_i} \subseteq \mathcal{O}_K[y]$ of degree at most l such that $\|m\| \leq B \Longrightarrow \|c(m)\| < F$.
3: Find all roots of c and report those roots ξ such that $\|\xi\| \leq B$ and $\sum_i a_i z_i > Z$.

VI. Existence of the decoding polynomial

In this section, given weights $(z_i)_{i\leq n}$, we prove the existence of a polynomial $c \in \prod_i J_i^{z_i}$ and a constant $F > 0$ such that for all $\|m\| \leq B$, $m \in \mathcal{O}_K$, we have $\|c(m)\| \leq F$. This proof is not constructive. The actual computation of this polynomial will be described in Section VII. We first need to estimate the number of elements of $\mathcal{O}_K$ bounded by a given size.

Lemma 1. *Let $F' > 0$ and $0 < \gamma < 1$, then the number of $x \in \mathcal{O}_K$ such that $\|x\| \leq F'$ is at least*

$$\left\lfloor \frac{\pi^{d/2} F'^d}{2^{r_1+r_2-1+\gamma}\sqrt{|\Delta|}\Gamma(d/2)} \right\rfloor.$$

Proof: As in [14, Chap. 5], we use the standard results of Minkowski theory for our purposes. More precisely, there is an isomorphism $f : K_{\mathbb{R}} \longrightarrow \mathbb{R}^{r_1+2r_2}$ and a scalar product $(x, y) := \sum_{i\leq r_1} x_i y_i + \sum_{r_1<i\leq r_1+2r_2} 2x_i y_i$ on $\mathbb{R}^{r_1+2r_2}$ transferring the canonical measure from $K_{\mathbb{R}}$ to $\mathbb{R}^{r_1+2r_2}$. Let $\lambda = f(\mathcal{O}_K)$, $X := \{x \in K_{\mathbb{R}} \mid \|x\| \leq F'\}$, and $m \in \mathbb{N}$. We know from Minkowski's lattice point theorem that if $\mathrm{Vol}(X) > m2^d \det(\lambda)$, then $\#(f(x) \cap \lambda) \geq m$. As $\mathrm{Vol}(X) = 2^{r_2}\left(2\pi^{d/2}F'^d/\Gamma(d/2)\right)$ and $\det(\lambda) = \sqrt{|\Delta|}$, we have the desired result. ∎

Then, we must derive from Lemma 1 an analogue of [9, Lemma 7.6] in our context. This lemma allows us to estimate the number of polynomials of degree l satisfying (2). To simplify the expressions, we use the following notation in the rest of the paper

$$\alpha_{d,\Delta,\gamma} := \frac{\pi^{d/2}}{2^{r_1+r_2-1+\gamma}\sqrt{|\Delta|}\Gamma(d/2)}.$$

Lemma 2. *For positive integers B, F', the number of polynomials $c \in \mathcal{O}_K[y]$ of degree at most l satisfying* (2) *is at least*

$$\left(\alpha_{d,\Delta,\gamma}\left(\frac{F'}{(l+1)B^{l/2}}\right)^d\right)^{l+1}.$$

Proof: Let $c(y) = c_0 + c_1 y + \cdots + c_l y^l$. We want the c_i's to satisfy $\|c_i m^i\| < F'/(l+1)$ whenever $\|m\| \leq B$. This is the case when $\|c_i\| < F'/(B^i(l+1))$. By Lemma 1, there are at least $\alpha_{d,\Delta,\gamma}\left(F'/((l+1)B^i)\right)^d$ possibilities for c_i. Therefore, the number of polynomials c satisfying (2) is at least

$$(\alpha_{d,\Delta,\gamma})^{l+1}\left(\left(\frac{F'}{l+1}\right)^{l+1}\prod_{i=0}^{l} B^{-i}\right)^d,$$

which finishes the proof. ∎

Now that we know how to estimate the number of $c \in \mathcal{O}_K[y]$ of degree at most l satisfying (2), we need to find a lower bound on F to ensure that we can find such a polynomial in $\prod_i J_i^{z_i}$. The following lemma is an equivalent of [9, Lemma 7.7].

Lemma 3. *Let l, B, F be positive integers, there exists $c \in \prod_i J_i^{z_i}$ satisfying* (2) *provided that*

$$F > 2(l+1)B^{l/2}\frac{1}{(\alpha_{d,\Delta,\gamma})^{1/d}}\left(\prod_i \mathcal{N}(\mathfrak{p}_i)^{\binom{z_i+1}{2}}\right)^{\frac{1}{d(l+1)}}. \tag{4}$$

Proof: Let us apply Lemma 2 to $F' = F/2$. There are at least

$$\left(\alpha_{d,\Delta,\gamma}\left(\frac{F/2}{(l+1)B^{l/2}}\right)^d\right)^{l+1}$$

polynomial $c \in \mathcal{O}_K[y]$ satisfying $\|m\| \leq B \Rightarrow \|c(m)\| < F/2$. In addition, we know from [9, Corollary 7.5] that $\prod_i |\mathcal{N}(\mathfrak{p}_i)|^{\binom{z_i+1}{2}} \geq |\mathcal{O}_K[y]/\prod_i J_i^{z_i}|$, which implies that if (4) is satisfied, then necessarily

$$\left(\alpha_{d,\Delta,\gamma}\left(\frac{F/2}{(l+1)B^{l/2}}\right)^d\right)^{l+1} > \left|\mathcal{O}_K[y]/\prod_i J_i^{z_i}\right|.$$

This means that there are at least two distinct polynomials $c_1, c_2 \in \mathcal{O}_K[y]$ of degree at most l such that $(c_1 - c_2) \in \prod_i J_i^{z_i}$ and $\|c_1(m)\|, \|c_2(m)\| < F/2$ whenever $\|m\| \leq B$. The choice of $c := c_1 - c_2$ finishes the proof. ∎

VII. Computation of the decoding polynomial

Let $l > 0$ be an integer to be determined later. To compute $c \in \prod_i J_i^{z_i}$ of degree at most l satisfying (2), we need to find a short pseudo-basis of the sub $\mathcal{O}_K$-module $M \cap \prod_i J_i^{z_i}$ of K^{l+1} where M is the $\mathcal{O}_K$-module of the elements of $\mathcal{O}_K[y]$ of degree at most l embedded in K^{l+1} via $\sum_i c_i y^i \to (c_i)$. We first compute a pseudo-generating set for each $M \cap J_i^{z_i}$, then we compute a pseudo-basis for their intersection, and we finally call the algorithm of [6] to produce a short pseudo-basis of $M \cap \prod_i J_i^{z_i}$ from which we derive c.

An algorithm for computing a pseudo-basis of the intersection of two modules given by their pseudo-basis is described by Cohen in [4, 1.5.2]. It relies on the HNF algorithm for $\mathcal{O}_K$-modules. The HNF algorithm presented in [4, 1.4] is not polynomial, but a variant recently presented in [2] enjoys this property. We can therefore apply [4, 1.5.2] with the HNF of [2] successively for each pseudo-basis of $M \cap J_i^{z_i}$ to produce a pseudo-basis of $M \cap \prod_i J_i^{z_i}$.

Algorithm 2 Computation of the decoding polynomial

Require: $(\mathfrak{p}_i, z_i)_{i \leq n}$, l, N, B, F such that $\exists c \in \prod_i J_i^{z_i}$ of degree at most l satisfying (2) for F, and the encoded message $(r_1, \cdots, r_n) \in \prod_i \mathcal{O}_K/\mathfrak{p}_i$.

Ensure: $c \in \prod_i J_i^{z_i}$ satisfying (2) for $F' = 2^{\frac{dl}{2}} \sqrt{l+1} \left(2^{2+d(6+3d)} d^3 |\Delta|^{2+\frac{11}{2d}}\right) F$ of degree at most l.

1: **for** $i \leq n$ **do**
2: $\tilde{z_i} \leftarrow \min(z_i, l)$.
3: For $0 \leq j \leq \tilde{z_i}$: $\mathfrak{a}_j^i \leftarrow \mathfrak{p}_i^{z_i - j}$, $a_j^i \leftarrow (y - r_i)^j$.
4: For $1 \leq j \leq l - z_i$: $\mathfrak{a}_j^i \leftarrow \mathcal{O}_K$, $a_j^i \leftarrow y^j (y - r_i)^{z_i}$.
5: Let $\left((a_j^i), (\mathfrak{a}_j^i)_{j \leq l+1}\right)$ be a pseudo matrix for $M \cap J_i^{z_i}$.
6: **end for**
7: Compute a pseudo-basis $[(c_i), (\mathfrak{c}_i)]_{i \leq l+1}$ of $M_1 = M \cap \prod_i J_i^{z_i}$.
8: Deduce a pseudo-basis $[(d_i), (\mathfrak{d}_i)]_{i \leq l+1}$ of the module M_2 given by

$$(v_0, v_1, \cdots, v_l) \in M_1 \Longleftrightarrow (v_0, v_1 \cdot B, \cdots, v_l \cdot (B)^l) \in M_2.$$

9: Let $[(b_i), (\mathfrak{b}_i)]_{i \leq l+1}$ be a short pseudo-basis of M_2 obtained with the reduction algorithm of [6].
10: Let x_1, x_2 be a short basis of $\mathfrak{b}_1$ obtained with [6, Th. 3].
11: **return** $c \in M_1$ corresponding to $x_1 b_1 \in M_2$.

VIII. Good weight settings

To derive our main result, we need to consider weights $z_i > 0$ in $\mathbb{R}$ rather than $\mathbb{Z}$. Let

$$\beta_{d,\Delta,\gamma} := \frac{d^{3-\frac{d}{2}} 2^{3(1+d(2+d))} |\Delta|^{2+\frac{11}{2d}}}{\alpha_{d,\Delta,\gamma}{}^{\frac{1}{d}}},$$

then by combining (3), (4) and Algorithm 2, we know that given $(r_1, \cdots, r_n) \in \prod_{i \leq n} \mathcal{O}_K/\mathfrak{p}_i$, $l > 0$, $B = \prod_{i \leq k} \mathcal{N}(\mathfrak{p}_i)^{1/d}$ and integer weights $z_i > 0$, Algorithm 2 returns a polynomial c of degree at most l such that all $m \in \mathcal{O}_K$ satisfying $\|m\| \leq B$ and

$$\sum_{i \leq n} a_i z_i \log \mathcal{N}(\mathfrak{p}_i) \geq \frac{l}{2} \log(2^{d^2} B^d) + \frac{3d}{2} \log(l+1) + \frac{1}{l+1} \sum_{i \leq n} \binom{z_i + 1}{2} \log \mathcal{N}(\mathfrak{p}_i) + \log \beta_{d,\Delta,\gamma}, \quad (5)$$

(where $a_i = 1$ if $m \bmod \mathfrak{p}_i = r_i$, 0 otherwise) are roots of c. In the following, we no longer assume the z_i to be integers. However, we will use our previous results with the integer weights $z_i^* := \lceil A z_i \rceil$ for a sufficiently large integer A to be determined.

Proposition 1. *Let $\varepsilon > 0$, non-negative reals z_i, $B = \prod_{i \leq k} \mathcal{N}(\mathfrak{p}_i)^{1/d}$, and an encoded message $(r_1, \cdots, r_n) \in \prod_i \mathcal{O}_K/\mathfrak{p}_i$, then our algorithm finds all the $m \in \mathcal{O}_K$ such that $\|m\| \leq B$ and*

$$\sum_{i \leq n} a_i z_i \log \mathcal{N}(\mathfrak{p}_i) \geq \sqrt{\log(2^{d^2} B^d) \left(\sum_{i \leq n} z_i^2 \log \mathcal{N}(\mathfrak{p}_i) + \varepsilon z_{max}^2 \right)},$$

where $a_i = 1$ if $m \bmod \mathfrak{p}_i = r_i$, 0 otherwise.

Proof: Note that we can assume without loss of generality that $z_{max} = 1$. Let $z_i^* = \lceil A z_i \rceil$ for a sufficiently large integer A, which thus satisfies $A z_i \leq z_i^* < A z_i + 1$. The decoding condition (5) is met whenever

$$\sum_{i \leq n} a_i z_i \log \mathcal{N}(\mathfrak{p}_i) \geq \frac{l}{2A} \log(2^{d^2} B^d) + \frac{3d}{2A} \log(l+1) + \frac{A}{2(l+1)} \sum_{i \leq n} \left(z_i^2 + \frac{3}{A} z_i + \frac{2}{A^2} \right) \log \mathcal{N}(\mathfrak{p}_i) + \frac{1}{A} \log \beta_{d,\Delta,\gamma}. \quad (6)$$

Let $Z_i := z_i^2 + \frac{3}{A} z_i + \frac{2}{A^2}$ for $i \leq n$ and

$$l := \left\lceil A \sqrt{\frac{\sum_{i \leq n} Z_i \log \mathcal{N}(\mathfrak{p}_i)}{\log(2^{d^2} B^d)}} \right\rceil - 1.$$

We assume that $A \geq \log(2^{d^2} B^d)$, which ensures that $l > 0$. For this choice of l, condition (6) is satisfied whenever

$$\sum_{i \leq n} a_i z_i \log \mathcal{N}(\mathfrak{p}_i) \geq \frac{3d}{2A} \log \left(A \sqrt{\frac{\sum_{i \leq n} Z_i \log \mathcal{N}(\mathfrak{p}_i)}{\log(2^{d^2} B^d)}} + 1 \right) + \sqrt{\log(2^{d^2} B^d) \left(\sum_{i \leq n} Z_i \log \mathcal{N}(\mathfrak{p}_i) \right)} + \frac{1}{A} \log \beta_{d,\Delta,\gamma}. \quad (7)$$

Assume that $A \geq \frac{10 \log N}{\varepsilon}$ and $A \geq \frac{\log \beta_{d,\Delta,\gamma}}{\log N}$, then for N large enough, the right side of (7) is at most

$$O\left(\frac{\log\log N}{\log N}\right) + \sqrt{\log(2^{d^2}B^d)\left(\sum_{i\leq n} z_i^2 \log \mathcal{N}(\mathfrak{p}_i) + \frac{\varepsilon}{2}\right)}$$

$$\leq \sqrt{\log(2^{d^2}B^d)\left(\sum_{i\leq n} z_i^2 \log \mathcal{N}(\mathfrak{p}_i) + \varepsilon\right)}$$

The degree l of our decoding polynomial c is therefore polynomial in $\log N$, $\frac{1}{\varepsilon}$, d and $\log|\Delta|$. By [1, 2.3], we know that the complexity to find the roots of c is polynomial in d, l and in the logarithm of the height of c, which we already proved to be polynomial in the desired values.

■

Corollary 1. *Let $\varepsilon > 0$, $k < n$ and prime ideals $\mathfrak{p}_1, \cdots \mathfrak{p}_n$ satisfying $\mathcal{N}(\mathfrak{p}_i) < \mathcal{N}(\mathfrak{p}_{i+1})$ and $\log \mathcal{N}(\mathfrak{p}_{k+1}) \geq (k \log \mathcal{N}(\mathfrak{p}_k) + d^2)$, then with the previous notations, our algorithm finds a list of all codewords which agree with a received word in t places provided $t \geq \sqrt{k(n+\varepsilon)}$.*

Proof: The proof is similar to the one of [9, Th. 7.14]. The main difference is that we define $\delta := k - \frac{\log(2^{d^2}B^d)}{\log \mathcal{N}(\mathfrak{p}_{k+1})}$ which satisfies $\delta \geq 0$ since by assumption $\log \mathcal{N}(\mathfrak{p}_{k+1}) \geq (k \log \mathcal{N}(\mathfrak{p}_k) + d^2)$. We apply Proposition 1 with $z_i = 1/\log \mathcal{N}(\mathfrak{p}_i)$ for $i \geq k+1$, $z_i = 1/\log \mathcal{N}(\mathfrak{p}_{k+1})$ for $i \leq k$, and $\varepsilon' = \varepsilon/\log \mathcal{N}(\mathfrak{p}_{k+1})$. It allows us to retrieve the codewords whose number of agreements t is at least

$$\sqrt{\frac{\log(2^{d^2}B^d)}{\log \mathcal{N}(\mathfrak{p}_{k+1})}\left(\frac{\log(B)}{\log \mathcal{N}(\mathfrak{p}_{k+1})} + \sum_{i=k+1}^{n} \frac{\mathcal{N}(\mathfrak{p}_{k+1})}{\log \mathcal{N}(\mathfrak{p}_i)} + \varepsilon'\right)}$$

$$\leq \delta + \sqrt{\frac{\log(2^{d^2}B^d)}{\log \mathcal{N}(\mathfrak{p}_{k+1})}\left(\frac{\log(2^{d^2}B^d)}{\log \mathcal{N}(\mathfrak{p}_{k+1})} + \sum_{i=k+1}^{n} \frac{\mathcal{N}(\mathfrak{p}_{k+1})}{\log \mathcal{N}(\mathfrak{p}_i)} + \varepsilon\right)}.$$

This condition is met whenever $t \geq \delta + \sqrt{(k-\delta)(n-\delta+\varepsilon)}$. From the Cauchy-Schwartz inequality, we notice that

$$\sqrt{k(n+\varepsilon)} \geq \sqrt{(k-\delta)(n-\delta+\varepsilon)},$$

which proves that our decoding algorithm works when $t \geq \sqrt{k(n+\varepsilon)}$. ■

IX. Conclusion

We presented the first method for list decoding number field codes. A straightforward application of Theorem 2 allows to derive a decoding algorithm in polynomial time. However, we cannot achieve the Johnson bound with this method. To solve this problem, we described an analogue of the CRT list decoding algorithm for codes based on number fields. This is the first algorithm allowing list decoding of number field codes up to the Johnson bound. We followed the approach of [9, Ch. 7] that provides a general frameworks for list decoding of algebraic codes, along with its application to CRT codes. The modifications to make this strategy efficient in the context of number fields are substantial. We needed to refer to the theory of modules over a Dedekind domain, and carefully analyse the process of intersecting them, as well as finding short elements. We proved that our algorithm is polynomial in the size of the input, that is in d, $\log(N)$, $\log|\Delta|$ and $\frac{1}{\varepsilon}$.

Acknowledgment

The first author would like to thank Guillaume Hanrot for his helpful comments on the approach based on Coppersmith's theorem. We also thank an anonymous referee for helpful comments on this paper.

References

[1] A. Ayad, "A lecture on the complexity of factoring polynomials over global fields," *International Mathematical Forum*, vol. 5, no. 10, pp. 477–486, 2010.

[2] J.-F. Biasse and C. Fieker, "A polynomial time algorithm for computing the hnf of a module over the integers of a number field," 2012, To appear in the proceedings of the 37th International Symposium on Symbolic and Algebraic Computation (ISSAC).

[3] D. Boneh, "Finding smooth integers in short intervals using crt decoding," in *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, ser. STOC '00. New York, NY, USA: ACM, 2000, pp. 265–272. [Online]. Available: http://doi.acm.org/10.1145/335305.335337

[4] H. Cohen, *Advanced topics in computational algebraic number theory*, ser. Graduate Texts in Mathematics. Springer-Verlag, 1991, vol. 193.

[5] H. Cohn and N. Heninger, "Ideal forms of coppersmith's theorem and guruswami-sudan list decoding," in *Proceedings of Innovations in computer science*, 2011.

[6] C. Fieker and D. Stehlé, "Short bases of lattices over number fields," in *Algorithmic Number Theory, 9th International Symposium, ANTS-IX, Nancy, France, July 19-23, 2010. Proceedings*, ser. Lecture Notes in Computer Science, G. Hanrot, F. Morain, and E. Thomé, Eds., vol. 6197. Springer, 2010, pp. 157–173.

[7] O. Goldreich, D. Ron, and M. Sudan, "Chinese remaindering with errors," in *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, ser. STOC '99. New York, NY, USA: ACM, 1999, pp. 225–234.

[8] V. Guruswami, "Constructions of codes from number fields," *IEEE Transactions on Information Theory*, vol. 49, no. 3, pp. 594–603, 2003.

[9] ——, *List Decoding of Error-Correcting Codes: Winning Thesis of the 2002 ACM Doctoral Dissertation Competition (Lecture Notes in Computer Science)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[10] V. Guruswami, A. Sahai, and M. Sudan, "Soft-decision decoding of chinese remainder codes," in *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*. Washington, DC, USA: IEEE Computer Society, 2000, pp. 159–168.

[11] V. Guruswami and M. Sudan, "Improved decoding of reed-solomon and algebraic-geometric codes," in *IEEE Symposium on Foundations of Computer Science*, vol. 5, 1999, pp. 28–39.

[12] H. Lenstra, "Codes from algebraic number fields," in *Mathematics and computer science II, Fundamental contributions in the Netherlands since 1945*, ser. CWI Monograph, M. Hazewinkel, J. Lenstra, and L. L. Meertens, Eds., vol. 4, North-Holland, Amsterdam, 1986, pp. 94–104.

[13] D. Mandelbaum, "On a class of arithmetic codes and a decoding algorithm (corresp.)," *IEEE Transactions on Information Theory*, vol. 22, pp. 85–88, 1976.

[14] J. Neukirch, *Algebraic number theory*, ser. Comprehensive Studies in Mathematics. Springer-Verlag, 1999, iSBN 3-540-65399-6.